DOCUMENT RESUME

ED 310 037 SO 020 134

AUTHOR Walbert, Mark S.
TITLE Avoiding Pedagogically Naive "Captive" Software.
PUB DATE 9 Apr 88
NOTE 18p.; Paper presented at the Midwest Economics Association Conference (Chicago, IL, April 7-9, 1988).
PUB TYPE Reports - Evaluative/Feasibility (142) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Computer Assisted Instruction; Computers; Computer Simulation; Computer Software; *Courseware; *Economics Education; *Instructional Effectiveness; Instructional Material Evaluation; Postsecondary Education; Textbooks

ABSTRACT

An increasing number of non-statistical software packages are being written as supplementary instructional materials provided free or at low cost for economics principles textbooks. This paper reviews the software programs currently available as ancillary material to several major texts and compares what is available as a group against what should be available if the goal is to use the microcomputer as an effective instructional tool. Much of the current crop of software does not effectively use the computer to instruct the student in a way only the computer can. Instead, much of the software that is written is largely an electronic version of the familiar student workbook. Often this places the student in the role of an electronic page turner. To remedy this shortfall, 14 specific modifications to the software designs are recommended including the following: (1) the software package should consist of menu-driven tutorials and drills such as model simulations, an electronic sketchpad, and a database spreadsheet; (2) the tutorial program should take the form of a dialog, in the Socratic sense; (3) questions in the drill portion of the package should automatically increase in difficulty so the student remains challenged by the questions; and (4) programs should not stand alone, but should be complemented with a brief student guide showing the student how to sign on, manipulate the cursor, and get help, as well as state the explicit instructional objectives for each of the programs in the package. (Author/PPB)

**********
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
**********

ED310037

# AVOIDING PEDAGOGICALLY NAIVE "CAPTIVE "SOFTWARE

Mark S. Walbert
Assistant Professor
Department of Economics
Illinois State University

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☐ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

MARK S. WALBERT

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

SO 020 134

Paper presented
at the
Midwest Economics Association Conference
April 7-9, 1989
Chicago, Illinois

# Avoiding Pedagogically Naïve "Captive" Software

Mark S. Walbert
Assistant Professor
Department of Economics
Illinois State University
(309) 438-7647

ABSTRACT

An increasing number of non-statistical software packages are being written as supplementary instructional material provided free (or at low cost) for economics Principles texts. This paper reviews the software programs currently available as ancillary material to several major Principles texts. This is not a product review per se, but a comparison of what is currently available as a group against what should be available if our goal is to use the microcomputer as an effective instructional tool. Much of the current crop of software does not effectively use the computer to instruct the student in a way only the computer can. Instead, much of the software that is written is largely an electronic version of the familiar student workbook. Often this places the student in the role of a mere "electronic page turner". To remedy this shortfall, a number of specific modifications to the software design are recommended.

Forthcoming, *Journal of Economic Education*, November 1989.
Presented at the Midwest Economics Association Conference
April 7-9, 1988, Chicago, IL.

# Avoiding Pedagogically Naïve "Captive" Software

Mark S. Walbert[1]

I hear and I forget;
I see and I remember;
I do and I understand.

## INTRODUCTION

As the old adage reminds us, learning economic concepts takes place only when the student is actively involved in learning economics, as opposed to passively listening to a lecture, or watching someone else work through a problem. During the 1970's, educators were enthusiastic about the benefits of using the University's mainframe computer to let the students "do" economics.[2] The hope was that computer-assisted instruction (CAI) could tailor economic education to take account of individual variations in students' abilities in a way the textbook and lecture approach could not. A large number of software programs were developed. Many of these programs were tested for their effect on the student's ability to understand economics, and the results were disappointing.

By the end of the decade Siegfried and Fels (1979) had written the obituary for CAI, noting that "Overall, games and CAI in economics do not appear to be the route to nirvana they were once expected to be. CAI appears to generate no more (or no less) cognitive achievement, but probably costs more than conventional pedagogical methods" (p. 942). The same problem existed in other disciplines and at other levels of education as well—the promise of CAI

---

[1]Assistant Professor, Department of Economics, Illinois State University. I am indebted to Alan Dillingham for his enthusiastic support and critical review of earlier drafts of this paper.

[2]See surveys by Soper (1974), and by Siegfried and Fels (1979, pages 940-944).

seemed unfulfilled. The consensus seemed to be that the computer was not the "dream" teaching tool as once thought.

But it's not the computer hardware that helps the student learn, it's the software—the lines of code that tell the computer how to respond to input from the user. Yet one finds little research in the literature that questions the quality of the software, or the way in which it was used. In their review of the failure of the promise of CAI, Schenk and Silvia (1984) concluded that one of the main reasons for the demise of mainframe CAI may have been the computer software, not the method. They note that evaluations of the effectiveness of CAI did not discuss the "possibility that the material itself may be of poor quality", or that good programs were improperly used (p. 240).

Now we are seeing a resurgence of interest in using the computer as an aid in the teaching of economics. The focus this time, however, is on the use of microcomputers to help students understand economics. This resurgence is taking the form of an increasing number of non-statistical software packages included as supplementary learning material provided free or at low cost for economics texts. Most of these "captive" software packages are directed at the principles level (where product differentiation is most crucial), but a few are written for higher level theory texts.[3] When one looks at the "captive" software, however, one quickly notices that few programs have been designed to really enhance and motivate learning.

---

[3]Examples of the software reviewed for this paper include software accompanying principles texts by McConnell, Byrnes and Stone, Baumol and Blinder, Truett and Truett, Dolan, Samuelson and Nordhaus, Ruffin and Gregory, Thompson, McKechearn and Amacher and Ulbrich; and software accompanying the more advanced texts by Friedman, Price Theory, Kohler, Intermediate Microeconomics, Dornbusch and Fisher, Intermediate Macroeconomics, and Keating and Wilson, Managerial Economics.

For science education, which moved into using micro's earlier, Klopfer (1986) reports finding the same problems with CAI on micros as on mainframes. He cites the "shortcomings of currently available science instructional software" as one important reason why the anticipated benefits from science instruction with microcomputers "has largely eluded us" (p. 16):

> The fact is that high quality instructional software for science represents only a small percent of the total. The great majority of the currently distributed software is disappointing in quality, pedagogically naïve and mundane (p. 17).

The very same can be said about the majority of the software packages that accompany principles texts today: they are "pedagogically naïve and mundane". The programs are often little more than electronic versions of the study guide, and the student's role is essentially reduced to turning electronic pages ("Press spacebar to continue").

The thesis of this article is that greater effort should be made to avoid setting microcomputer-aided instruction down the same dead-end path as mainframe CAI. In particular, more work is needed in the *design* of software from the point of how to best affect the students' abilities to learn. To date, very few of the software packages accompanying principles texts are structured in a way that uses the computer as an instructional tool to do what can only be done on a computer[4].

This article examines the types of software programs currently available with most introductory economics texts and compares what is currently available against what should be available if our goal is to use the microcomputer ef-

---

[4]For a more detailed discussion of the problems specific to the programs serving as the basis for this paper, see Walbert (1988).

fectively to aid the student's ability to learn economic concepts. This is not intended to be a product review[5], because such a review would simply be a catalog, unless a framework for evaluation is first developed. The two goals of the paper are, first, to address how well the current crop of "captive" software uses the microcomputer as a learning tool, and second, to discuss what changes should be made to improve the current software design. To put the article in perspective, the paper begins with a discussion of why the modern microcomputer should be given consideration as a significant learning tool.

## LEARNING WITH COMPUTERS

Microcomputers are getting less and less expensive, as they get more and more powerful. As a result, an increasing number of departments are able to build microcomputer "labs" where students can be exposed to a number of different computer operating systems (i.e., Apple, MS-DOS, Macintosh, etc.). Processing time is falling to the point where, for many applications, the turnaround time is shorter than on a multi-user mainframe (especially for printouts of results). Internal memory and hard disk storage capacity is increasing. Compiling time for many programming languages has been reduced as well. As a result, more complicated software packages can be written because the average machine will have the necessary internal memory needed to download the entire program into RAM. This will significantly reduce the time it takes to run the program, and that will help maintain student interest in using it.

These advances in microcomputer technology mean the machine no longer presents much of a constraint on what we can do with educational software. But

---

[5]Such a review has been recently undertaken by Yoho (1988), who coined the term "captive" software.

it's still properly written software that makes the computer an effective instructional complement to the current lecture/workbook approach to promoting student learning[6].

With well written software, the microcomputer can help promote learning in at least five ways. First, it can serve as a patient personal tutor; endlessly reviewing concepts, graphs, and mathematical models that are crucial, though difficult, for the student to understand. Second, it can serve as a drill-master, repeatedly giving the student instant feedback on incorrect (or correct) answers to practice problems or pre-exam drills over a number of variations on a single learning theme. Third, it can offer the student a "what if...?" learning environment with instantaneous feedback to encourage development of a student's economic intuition through independent exploration of the mechanics of economic models. Fourth, it can monitor the student's understanding of graphical models my means of an electronic sketchpad. Fifth, it can keep a record of the student's progress, and suggest specific readings for additional study, additional workbook problems for more practice, or one-on-one sessions with the instructor. Repetitious practice, positive reinforcement, directed guidance, and an environment encouraging independent exploration are very important ways to promote involvement in learning, but very tedious for the human tutor. Used in this way, microcomputer software packages can help "reprogram" the way a student learns from the standard model of learning that says one either "gets it"

---

[6]These points are also made by Lovell (1987) in a paper which examines the opportunities for using CAI on personal computers in the <u>teaching</u> of economics. Note also his comments on the market distribution problems posed by the public good nature of software.

or one "gets it wrong", to a more appropriate model that says learning is a *process*, a method of "debugging" one's way of thinking[7].

By using the microcomputer software as a personal tutor, a drill-master, an electronic sketchpad, and a "what if...?" environment, the student can get instant feedback on the degree of progress he or she is making in understanding the lessons taught in lecture. The student will be able to bring the computer's diagnostics to the professor so they may focus on the one or two topics giving the student the greatest difficulty. In tandem with instructor assistance and feedback, the programs can help take much of the fear out of being wrong that serves to inhibit much of the student's learning.

## CURRENT USES OF THE MICROCOMPUTER IN PRINCIPLES INSTRUCTION

Pedagogically sound microcomputer software offers something for the educator no other medium does: it can help involve the student in the learning process, and help the student learn to think independently. This is just what the doctor ordered: all the reports on the poor state of higher education cite a failure to involve students in the learning process, and a failure to promote independent thinking. Unfortunately, most of the "captive" software packages do neither. This section briefly outlines the current use of "captive" software highlighting the aspects that do not contribute to the computer's ability to do more than a written text. The next section lists ways to improve on the pedagogical quality of such software.

---

[7]This is a point argued by Pappert (1980, p. 23), based on concepts of learning explored by Piaget.

The many programs currently available with principles texts may be classified into three types: electronic workbooks; simulations and games; and databased spreadsheets.

The type of program found in one form or another on almost all the "captive" software is the electronic workbook. At best the electronic workbook contains either a tutorial program, a drill/pre-test program, or both (as in the software accompanying Thompson, and to a small extent that accompanying Baumol and Blinder). At worst the program is merely an electronic duplicate of the paperback student Study Guide accompanying the text. An example of this type is the software accompanying Bronfenbrenner, Sichel, and Gardner: the only difference between the software and the paperback Study Guide is the inclusion of a search procedure in the electronic version for locating terms in the glossary.

The tutorials are written along the lines of a monolog, with the student periodically asked questions to break the monotony. In only a few cases are the questions anything more than a means to nudge the drowsy student. Tutorials vary widely in terms of screen design, quality and speed of graphics, presence or absence of animation or sound, the availability of on-line help, difficulty of questions asked, statement of learning objectives, and the degree of post-tutorial diagnostics. Most are not as easy to "page through" as a paperback workbook; only the tutorial software accompanying McConnell's text allows the student to "page back" through the monolog.

The drills generally include a number of (usually multiple choice) questions (with Ruffin's and Gregory's software offering 1000 variations on each type of question), with the level of difficulty either unvarying or unspecified. They differ in terms of the quality of response given to incorrect answers (from one short sentence to paragraph-long responses), with no "captive" programs offering responses to correct answers. Some programs allow the student several attempts

at answering the question (even to the point of giving three hints when only four responses are possible), but do not allow them to skip a question and return to it later. No program systematically varies the level of problem difficulty,[8] nor does any program come with notes to the instructor indicating what different levels of questions are being asked.

A second type of program contains simulations or games. Simulations of the national economy were one of the first uses of CAI. There are few simulations currently available with principles texts[9]. Those that are available are written in closed box format (that is, neither the student nor the teacher may alter the underlying macroeconomic model). In macro-oriented programs, the student is charged with suggesting macroeconomic policies for dealing with a variety of economics ills. In the micro-oriented program, the student is charged with maximizing the profits of a firm under a variety of market responses. The value of simulations is in its ability to create an artificial economic world in which the student can experiment with responses to changes in economic stimuli. The primary problem with the current crop of closed-box simulation programs is the student is not able to alter the parameters of the underlying model to explore how it changes its structure; this removes the model itself from being an integral part of the learning process[10]. The underlying economic model is also not built

---

[8]The one program that does vary the level of problem difficulty, Market Master, is not tied to any specific principles text.

[9]Two are "MacroSim II" and "Raiders" by Byrnes and Stone. The new edition of "CAPER" (accompanying Economics by Baumol and Blinder) will have a macro-simulation model, but is not available for review at this writing. A non-text-specific macro model that avoids many of the problems discussed below, has been recently discussed in Case and Fair (1985).

[10]One could argue that model-*building* is best accomplished in a databased spreadsheet. Hence its absence in the current crop of simulation programs does not represent a problem with simulations *per se*.

up in stages as the student's level of understanding increases, but is presented to the student in complete form. Thus the student must wait until nearly the end of the semester to use the program to examine the possible policy options that have been discussed in class. The simulations currently available do make good use of tables and graphs to convey information on past policy decisions and results.

The third class of programs are databased spreadsheets. Databased spreadsheets are currently available with only two principles texts[11]. The programs are essentially spreadsheet templates (requiring the use of a spreadsheet package, such as Lotus 1-2-3) that contain actual data on unemployment, interest rates, etc., that allow the student to discover the relationships among important economic variables via simple mathematical calculations they make. With the ability to update the data, the information used by the student can remain current. Also, the mathematical skill level required of the student can be adjusted to correspond to the classes skill level and location in the course. The primary advantage offered by spreadsheets is the "what if...?" learning environment they present to the student. The current crop of "captive" spreadsheet templates limit the student to using data to verify and interpret the relationships between variables that are discussed in the text. They do not begin to tap the spreadsheet's potential for stimulating exploration of the mechanics of the economic model they are learning.

## SUGGESTIONS FOR MODIFICATION TO SOFTWARE DESIGN

In keeping with the view that the software needs redesigning in order to enhance and motivate the learning process, several pedagogically-motivated

---

[11]These are the DRI Data Diskettes supplied with the McGraw-Hill Economics texts by Samuelson and Nordhaus, and by McConnell.

suggestions for improving the educational potential of these programs are offered. Each of these suggestions is designed to promote software designs that utilize the computer's special abilities to calculating numbers quickly, provide clear animated graphics, and give the student instant, detailed feedback on her understanding of specific models and concepts.

The first suggestion is a general one:

1. At a minimum, the software package should include menu-driven tutorials *and* drills. But a package is complete only if it includes menu-driven open-model simulations, an electronic sketchpad, and a menu-driven databased spreadsheet (no order implied).

All five types of programs have the potential to promote learning, but financial considerations may restrict the size of the software package. In such a case, tutorials and directed drills can cover more of the material in a textbook than either simulations or databased spreadsheets. But spreadsheets and simulations promote a level of student involvement in learning that is much "higher" than tutorials or drills. They can also be used by instructors in the classroom; thus enabling them to integrate the software into lecture notes, assignments, and tests, and thereby getting the students more involved in learning economics and more proficient at thinking independently about economic issues[12].

One of the most important aspects of CAI is promotion of active student involvement with the computer package. Programs that move the study guide onto the screen, requiring the student to simply "turn" electronic pages are a waste of computer resources. Even programs using fairly decent animated

[12]My thanks to an anonymous reviewer for bringing out this point.

graphics fall short of their educational potential if the student does not *actively* relate to the computer. This leads to the following suggestions:

2. The tutorial program should take the form of a dialog, in the Socratic sense. In this case, the student would respond to a set of carefully planned questions leading to the lesson to be learned, rather than sit passively "turning" electronic pages. At second best, the tutorial would present pages of monolog, but would vary the difficulty of the material as a result of the student's response to questions.
3. Tutorials should screen up and down to allow review of previous dialog.
4. As a way to promote the interactive abilities of the computer, the electronic workbook should include an environment wherein the student can respond to graphical questions by using the screen as an electronic sketchpad—moving the cursor to an appropriate location on the graph to locate an equilibrium, draw a budget line, indicate curve shifts, maxima and minima, etc.[13] Ideally a joystick or mouse would be used to move the pointer across the screen but cursor keys may mimic their movements, though not nearly as well, and are already available on micro's. Ideally this would be part of the tutorial but, as a second best approach, it could be a separate program entirely.

Immediate feedback on responses to dialog questions, answers to drill problems, or changes to spreadsheets is another important aid to learning. This is where use of the computer can be very helpful; instead of relying only on a single weekly assignment or on a monthly exam to reveal his or her progress, the

---

[13]The software written by David Friedman to accompany his intermediate level Price Theory text is the first to apply this "electronic sketchpad" concept in a "captive" program.

student can get feedback each day by repeated reviewing of the relevant tutorial, drill, spreadsheet, or simulation. Thus:

5. Drills should give the student instant feedback on *right* as well as wrong answers. If a graph, table or equation is used in the question, it should be included in the explanation of the answer;
6. Tutorials should be able to respond to "how to" questions; how to determine price elasticity of demand, how to show an increase in supply, and so on, either with the answer or with the relevant pages in the textbook;
7. Spreadsheets should allow the student to view and vary the underlying models' parameters, and then to instantly see the results of any parameter changes graphically as well as in the tabular form of the spreadsheet.

One way to push the student to increase his or her level of performance is by varying the level of problem difficulty in the drill section of the package. This can be done easily on a computer, and in such a way as to make it rewarding for the student to attempt to consistently answer more difficult questions. At the same time, it allows the student to work at her best level, no matter what that level is. This includes giving the student the chance to "mull over" a drill question before typing in an answer. To this end, the following suggestions:

8. Questions in the drill portion of the package should automatically increase in difficulty so the student remains challenged by the questions. For example, the first level of questions may test only recognition; the second level would ask analytical questions (e.g., questions using graphs, tables, & equations); and the third level may present the student with synthesis questions, which seek to tie together new material with old;
9. Drills should also allow the student to skip a question when first asked, returning to it later.

Economics instruction at the principles level makes extensive use of graphical models. Thus the program must use high quality graphics and pay particular attention to their development. Thus:

10. Tutorials should make use of high quality graphics (requiring a graphics board on IBM-type machines), must draw slowly upon initial exposure, but re-draw quickly after changes have been made, and use animation or sound when illustrating shifts, etc;
11. Spreadsheets must include graphs and, where a model's parameter values have been changed, the graph should show the student the before and after change versions of the model.

Other effective learning techniques include clearly stating the learning objectives at the beginning of the program, and reviewing those objectives at the end, and asking questions or showing curiosity-provoking ambiguous information before the lesson begins.[14] Thus:

12. The program should be text specific. This is so the diagnostics at the end of the tutorial or drill program can inform the student of specific problem areas, and direct the student to *specific* pages, graphs, and tables in the text, and not just general chapter headings chapters, for further review; and so the terminology, model parameter notation, graphical models, and economic concepts are consistent among the tutorial, drill, simulation, spreadsheet, and text.

[14]For more strategies based on relevant psychological research findings, see MacLachlan (1988).

Finally, the software package should not be viewed as stand-alone teaching material; it is important for the student to have access to a human tutor who can offer additional help with the lesson or drill, offer suggestions on how to utilize the "what if...?" nature of the spreadsheet, respond to questions on the outcome of policy decisions in the simulation, and so on. This leads to two final suggestions:

13. Tutorials, drills, sketchpads, spreadsheets, and simulations should be created as a supplement only, not designed as a substitute for lecture or for reading the text;
14. Programs should not stand alone, but should be complemented with a brief brief student guide (or easy to use on-line help) showing the student how to sign on, manipulate the cursor, get help, etc., as well as stating the explicit instructional objectives for each of the programs in the package.

## CONCLUSION

Research on the effectiveness of mainframe CAI in economics has never asked what software ought to do in a learning context. This is an important issue, because it may have been the reason behind the demise of mainframe CAI. Now we have the opportunity to use CAI again, this time on a more decentralized machine, the microcomputer. Micro's are rapidly becoming as powerful as mainframes, and at a fraction of their cost. But so far, the software accompanying new principles texts has apparently not addressed the important question: what kind of software really motivates students, and enhances the learning process?

This paper recommends fourteen specific changes to the design of "captive" software that step in the direction of making the programs pedagogically sophisticated. Only when more instructionally focused software becomes available

for use with principles texts, can we begin to test whether micro-based CAI is effective in helping students learn economic principles.

## REFERENCES

Case, K.E., and R.C. Fair, "Macro Simulations for PCs in the Classroom", *American Economic Review*, May 1985, 85-90.

Klopfer, L.E., "Intelligent Tutoring Systems in Science Education: The Coming Generation of Computer-Based Instructional Programs", *Journal of Computers in Mathematics and Science Teaching*, Summer 1986, 16-32.

MacLachlan, J., "Psychologically Based Techniques for Improving Learning Within Computerized Tutorials", *Journal of Computer-Based Instruction*, Summer 1986, 65-70.

Pappert, S., *Mindstorms*, New York: Basic Books, 1980.

Schenk, R. and J.E., Silvia, "Why Has CAI Not Been More Successful in Economic Education: A Note", *Journal of Economic Education*, Summer 1984, 239-242.

Scherga, J.D., "Instruction in Economics Through Simulated Computer Programming", *Journal of Economic Education*, Spring 1986, 129-139.

Siegfried, J. and R. Fels, "Research on Teaching College Economics: A Survey". *Journal of Economic Literature*, September 1979, 923-969.

Soper, J.C., "Computer-Assisted Instruction in Economics: A Survey", *Journal of Economic Education*, Fall 1974, 5-29.

Walbert, M.S., "A Review of Microcomputer Software Programs Accompanying Selected Principles Texts", ~~forthcoming~~, *Economic Inquiry*, January 1989 ~~October 1988~~.

Yoho, D.L., "'Captive' Microcomputer Software for Economic Instruction: A Review", *Social Science Microcomputer Review*, Winter 1986, 425-437.